

Association for Information Systems AIS Electronic Library (AISeL)

International Research Workshop on IT Project
Management 2007

International Research Workshop on IT Project
Management (IRWITPM)

December 2007

Managing User Expectations on Software Projects: Lessons from the Trenches

Stacie Petter

University of Nebraska at Omaha

Follow this and additional works at: <http://aisel.aisnet.org/irwitpm2007>

Recommended Citation

Petter, Stacie, "Managing User Expectations on Software Projects: Lessons from the Trenches" (2007). *International Research Workshop on IT Project Management 2007*. 7.
<http://aisel.aisnet.org/irwitpm2007/7>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2007 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Managing User Expectations on Software Projects: Lessons from the Trenches

Stacie Petter

University of Nebraska at Omaha

spetter@mail.unomaha.edu

ABSTRACT

A common risk on software projects is managing user expectations. Software project managers have the difficult job of balancing the needs and requirements of the users with the time and budget allocated to the project. If a software project is over-promised and/or under-delivered, then the users will not be as satisfied with the system, thus affecting its future use. In this paper, I explain why it is necessary to manage user expectations on software projects by exploring the relevant literature. Second, I report the findings from a series of interviews conducted with software project managers that asked for challenging situations in which user expectations were successfully and not-as-successfully managed. These findings identify practical advice for software project managers on how to manage user expectations and offer insights to researchers for future research on software project management and managing user expectations.

Keywords

User expectations, vendor-client relations, software projects.

INTRODUCTION

Software projects are fraught with risks, with many risks appearing on all projects. In an attempt to identify software project risks and to determine which risks are more deserving of a project manager's attention, Schmidt et al. (2001) conducted a Delphi study with software project managers. Of the fifty-three risk factors identified by three different panels, eleven risk factors were consistent across the panels. Eight risk factors are outside of the software project manager's full control. Yet, three risk factors identified by the Delphi panels are within the software project manager's realm of control: failure to manage end user expectations, misunderstanding requirements, and insufficient/inappropriate staffing.

While the research by Schmidt et al. was useful in identifying risks, the authors left it to future research to determine how to address these risks during a software project. Therefore, to pick up where this prior research ended, I opted to explore one of these risk factors in depth. This research considers the following question:

What strategies and tactics can be utilized by software project managers to address the risk "failure to manage end user expectations" on a software project?

To answer this research question, I conducted interviews with software project managers to discover the lessons they have learned throughout their experience in terms of managing user expectations. I provide a framework as well as specific advice that can be used by software project managers to counter the risk of failing to manage user expectations.

LITERATURE REVIEW

Motivation

Expectations determine "the baseline or reference level for consumers to form evaluative judgments about the focal product or service" (Bhattacharjee 2001, p. 354). Managing user expectations is the actions of a software project manager to ensure that the assumptions held by the user for a software project are realistic and consistent with the software deliverable promised by the project team (Baccarini et al., 2004, Ginzberg, 1981). These expectations "must be correctly identified and constantly reinforced in order to avoid failure" (Schmidt et al., 2001, p. 15).

The Project Management Institute has stated that meeting user expectations is one of the primary criteria for project success (Project Management Institute, 2004). Furthermore, in a study of managing risks for IT projects, Baccarini et al. (2004) found that IT project managers rated "unrealistic expectations" as the third highest ranked project risk.

These surveys of software project managers conducted by Schmidt et al. (2001) and Baccarini et al. (2004) are consistent with Ginzberg's (1981) work on pre-implementation expectations of users. Ginzberg found that users with appropriate expectations about an information system had higher levels of use and user satisfaction than those with less realistic expectations. Ginzberg argued that "realism of expectations was a better predictor of success" than other measures used in his study.

There is significant research that user satisfaction affects the use and resulting impact of an information system (e.g., Bokhari 2005; DeLone and McLean 1992; DeLone and McLean 2003). Mahmood et al. (2000) performed a meta-analysis on several antecedents to user satisfaction of information systems. In their research, the authors identified seven studies that examined the relationship between user expectations and user satisfaction. The result of the meta-analysis determined that there was a significant relationship (effect size = 0.458) between user expectations and user satisfaction.

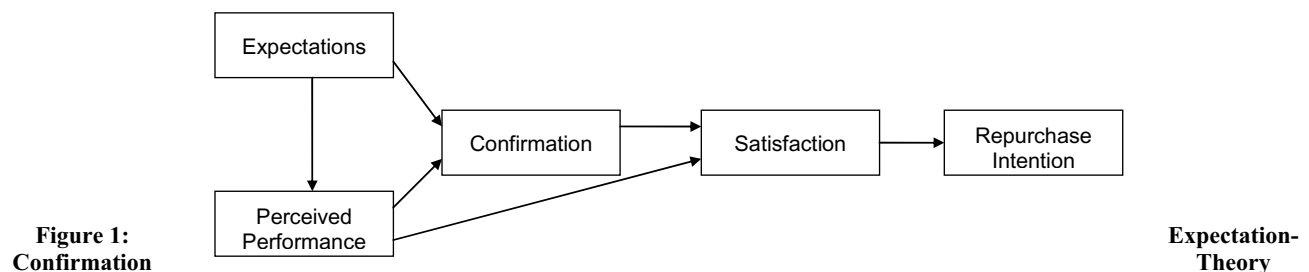
In a series of longitudinal experiments, Davis and Venkatesh (2004) determined that users develop their opinions and expectations of a system prior to hands-on use of the software. User opinions of Perceived Ease of Use, Perceived Usefulness, and Intention to Use were formed during the prototyping stage and remained consistent even after the users had an opportunity to use the system. This study confirms that it is critical to properly manage user expectations, even during early stages of the project, such as requirements analysis and design. Failing to develop appropriate and realistic expectations during software development, can affect future use and satisfaction of the system.

Related Theories

Two theories from the marketing literature have direct relevance to the phenomenon of managing user expectations: Expectation Confirmation Theory and SERVQUAL. Both theories have been used within the information systems literature and confirm the importance of managing user expectations within a software project.

Expectation Confirmation Theory

Expectation-confirmation theory is widely cited in the marketing literature to examine and understand consumer behavior (Bhattacharjee 2001). This theory suggests that individuals first develop an expectation of a product or service (or software, in this case) prior to its use. After using the product or service, the individual develops opinions regarding its performance. The performance of the product or service is compared to the individual's expectations in confirmation. Based on the confirmation level, the individual has a specific level of satisfaction, which then informs the consumer's repurchase intention. Figure 1 illustrates this framework as posited by Oliver (1980).



Expectation-confirmation theory has been applied to the IS field and the continued use of an information system. User satisfaction affects the continuance of usage of an information system (Bhattacharjee 2001); therefore, initial user expectations must be reasonable to ensure higher levels of satisfaction and thus higher levels of usage (Ginzberg 1981; Davis and Venkatesh 2004). Staples et al. (2002) study of librarians pre- and post-implementation of new cataloging software revealed that individuals with higher expectations of the software pre-implementation were less satisfied with the software post-implementation than individuals with lower expectations prior to the software's implementation. These findings illustrate the importance of not over-selling software during a project, which can result in lower levels of satisfaction, thus having the potential to affect the continued use of the system.

SERVQUAL

Another theoretical base to understand consumer expectations is SERVQUAL. This concept specifically examines consumer expectations and perceptions of a service. The difference between a person's expectations and perceptions, the gap, suggests where the firm is over-promising and/or under-delivering the service (Parasuraman et al. 1988). SERVQUAL has been adapted for the information systems literature to examine the service quality of technology departments within an organization (Kettinger and Lee 1995; Pitt et al. 1995). One function of many technology departments is to develop and create software. Software project management is not simply product development (Tiwana and Keil 2004), but is a combination of product and service delivery to offer a solution to the user. Kettinger and Lee (1994) found that IS service quality is related to satisfaction in that the higher the gap between expectations and perceptions, the lower the level of satisfaction. Therefore, the SERVQUAL literature reiterates the importance of software project managers to ensure users have reasonable expectations.

Both expectation-confirmation theory and SERVQUAL explain why it is important for software project managers to be concerned with managing user expectations, yet this literature does little to explain how a project manager can address this risk. Our research addresses this shortcoming.

RESEARCH STUDY

To examine the research question, I took an inductive approach to identify the strategies and tactics that can be used by software project managers to manage user expectations on a software project. Another researcher and I interviewed software project managers from the same industry sector in a large, IT and management consulting company of over 75,000 employees across nearly 50 countries. We selected twelve informants with a range of project management experience. Table 1 lists the informants along with job titles, years at that job title, and years of experience in project management.

Name ⁵	Job Title	Time in Position	Project Management Experience
Abigail	Consultant	2.5 years	3 years
Ryan	Consultant	3.5 years	3 years
Emily	Manager	.5 years	3 years
Anthony	Manager	.5 years	3 years
Hannah	Manager	1.5 years	3 years
Joseph	Manager	3 years	5 years
Christopher	Manager	1 year	6 years
Jacob	Manager	2 years	7 years
Joshua	Manager	3 years	7 years
Matthew	Manager	3 years	7 years
Ethan	Senior Manager	2 years	7 years
Andrew	Senior Manager	1.5 years	20 years

Table 1: Description of Informants

The interviews consisted of semi-structured questions related to challenges in managing user expectations, developing solutions to address the problem, and the results of the actions taken. We asked each software project manager to recall two situations in which they faced challenges in managing user expectations: one situation in which the software project manager believed the end result was successful and the other situation, in which looking back via hindsight, was not as successful. This allowed us to obtain data on twenty-four different situations in which the software project manager had to manage user expectations. After exhausting our questions, we offered the informant the opportunity to provide other insights regarding managing user expectations.

All interviews were conducted via telephone. Two researchers were present at each interview and took turns asking questions. Each researcher recorded personal notes and tape-recorded each interview to use for the analysis. Each interview was transcribed and yielded over 150 pages of text. Anonymity and confidentiality was assured to all informants.

Data Analysis

The situations (successful and not-as-successful), actions taken, results, and lessons learned were summarized in a table to compare the information across each mini-case study. This resulted in two tables of twelve cases each. The actions taken by each software project manager to manage user expectations were then summarized and categorized into a framework. Fifteen tactics for managing user expectations were implemented in the successful cases. These tactics were grouped into three general strategies to manage user expectations. I then examined the less-successful mini-cases and found nine unsuccessful tactics that also could be grouped within the three general strategies.

⁵ All names have been changed for confidentiality.

I compared the strategies used in both the successful and not-as-successful situations. This cross-check allowed us to establish additional credibility for our framework by confirming that an absence of one or more strategies led to a less-than-successful attempt at managing user expectations. As a final check of the results, after creating the framework, I compared the strategies to the text of each transcript to ensure the findings were consistent with the comments made by each informant.

Results

Three broad strategies to manage user expectations emerged from the analysis. For each of the situations that had a positive outcome, one or more of strategies were used to ensure that the users' expectations were managed appropriately. Situations with a negative outcome suffered from problems in one or more of the strategies. Table 2 lists the strategies and tactics identified in the research study.

General Strategy	Successful Tactics	Less Successful Tactics
User Involvement	Listening to users Asking questions Giving credit to specific users for ideas within the group Creating small groups for large projects to allow all to be heard Letting users make tough choices about budget, schedule, and/or functionality Understanding users' concerns regarding change and help them to feel at ease Keeping users involved throughout the project Working <i>with</i> users (not <i>at</i> them or <i>to</i> them)	Planning to "outlast" a difficult user rather than working with them Not communicating with users on the state of the project
Leadership	Educating users on the value and benefits of the system Obtaining buy-in from the primary (and/or most vocal stakeholders) and work outward Motivating the project team to get the project done Ensuring a strong project champion is sharing the vision Articulating a clear vision of the project	Failing to explain the purpose of the system Following others when you believe their actions are misguided Poor project sponsors (such as allowing politicking or other issues to affect the project) Failing to control for scope creep
Trust	Using clear terminology Sharing good news and bad news throughout the project	"Fake it until you make it" (i.e., not disclosing a lack of knowledge in an area) Hiding the true state of the project from the users

Table 2: A Framework for Managing User Expectations

User Involvement/Participation

The most common theme that occurred within our interviews was the importance of working with the users. Many software project managers recalled spending a lot of time talking with the users to understand their needs for the software. More specifically, the "talking" that occurred with the user focused more on listening to the user's concerns and asking questions.

It was more just an effort in learning how to listen...so I let him talk about what he worked on, what his area was, what his experiences were, why he was the person and the only person that knew it. (Jacob)

Many of the situations described by the software project managers were related to users facing a new process for doing their jobs as well as a new technology. Many users were resistant to this change and were hesitant to buy-in to the project. It was important for the users to understand that there was a sense of "we are all in this together" to be put at ease.

We were able to start develop a dialogue relationship with the user group so they would understand that this is something we were going to do with them and not to them. (Ethan)

For those working with large groups within a project, one tactic that was particularly helpful was breaking a large group into smaller groups during to allow individuals to speak, get issues off their chest, and then report findings back to the larger group. Another tactic was the importance of building positive momentum toward the project, and continuing it throughout the development phase. As implementation progresses, the importance of training, help desks, and other support functions became important to ensure that users were comfortable and felt involved as their work process changed.

Overall, communication with users was key. It was important that throughout the project, the users were kept informed of the progress of the project and what it was they were really getting.

The hurdle, as in all software development projects, is getting them involved in the various stages of the lifecycle of that project. So first thing is ensuring that you have correctly documented their requirements, and once you've completed the design, you go back to them and say okay, this is what we discussed, this is what it looks like, this is what it feels like, is that an appropriate solution for you. Once you have that buy-in, then you go back and complete the development, and you go back again before system testing, and once again provide them with the solution and what it's going to do and what the value is to them in the end and get their buy-in once again. Then you go into user acceptance testing...and essentially you've brought them along through the whole system development life cycle with you. (Andrew)

In the situations where there was a failure in managing user expectations, there was a lack of user involvement. In one situation, the users were not updated regularly on the software, the features, and plan for implementation. In this scenario, the resulting software had too many features that users did not understand, the software was a radical departure from their current process, and the users were less than satisfied. Upon reflecting on this project, a software project manager commented:

A lot of times, you say here's our prototype, in six months we'll give you this product. Well, the reality is the prototype is about forty percent right...Rather than sending an email and saying we have to change this and that, walk them through the change every time or on a regular basis...If you have a six month development process, then every two weeks you say here's where we are and what it looks like. By the way, here are some notes about the changes taking place. Then you get the opportunity to comment at that point in time, and everyone is on the same page regarding expectations and at the end of the day, the client doesn't feel like they're getting [something] different than what they expected. (Ryan)

Leadership

There are two types of leadership that need to be exhibited during a software project to properly manage user expectations: a project champion for the users and a project manager/leader for the team. The project champion helps to manage expectations by promoting the vision for the project, educating users on the value and benefits of the software, by rallying the "troops" around a common purpose, and explaining to the users how they can assist in the effort. The project managers wanted, and often needed, a representative from the user group that understood the value of the software and could educate other users. They needed someone that understood the benefits of the system and could focus the users on the positive changes that would be forthcoming. There was a need for at least one person within the user group that had a clear vision of the software to articulate the requirements and priorities of the system.

The software project managers offered several "lessons from the trenches" regarding what makes a good project champion. One project manager considered the type of champion she would want on a project.

I think you need a person who's well-respected within the organization. Someone who is influential, someone who has connections. Someone who has really good people skills who can help people to understand we are not threatening them. (Hannah)

Hannah, as well as other software project managers, realized the importance of having a champion that can put users at ease during a time of change. Several informants also discussed the importance of having a project champion that can encourage the users to put in the time required to develop requirements and review deliverables.

Other ideal qualities in a project champion included strong, clear goals for the project. One project described by Ryan appeared to be set up to fail due to aggressive timelines, inexperienced developers on the project team, and high visibility in the organization. However, the project champions had very clear goals for the project and prioritized their requests to allow the project team to focus on the most critical functionality. This strong leadership from the user group was certainly helpful in ensuring this high-risk project could be successful.

The second type of leadership critical to managing user expectations is a strong software project manager. The software project manager not only is responsible for making sure his/her team is getting work done effectively, but also for leading the users along the right path. Matthew described a situation in which he entered a project that was under-scoped and with only one resource on the project, an inexperienced developer. Sheer willpower was the reason for success for this project. Matthew taught the developer at night so she could work on the project during the day. He did not ask his team member to do more than he was willing to do.

The software project manager also needs to be a strong leader with the client. To do this, the software project manager needs knowledge about the business problem, the technical aspects of the system, and project management. Jacob recalled one of his first projects where he had little understanding of the business domain and the technical aspect of the system. He remembered thinking that “confidence is competence” and that at the end of the day he would simply “fake it until you make it.” He did not feel comfortable asking for help and did not want to be perceived as incompetent, regardless if it was true or not. Finally, a more senior manager stepped in and realized how much he was struggling with the project and brought in a third-party to help with the situation. The end result meant that there was a need for re-work, costing both time and money. Reflecting on the experience, Jacob believes now that it is important to ask for help when necessary as well as to know your personnel well enough to realize if they have the necessary knowledge to manage a given project.

If the software project manager has knowledge of an area, then it is important to illustrate that to the users. The client needs to know that the project manager understands the problem and has competence in the area. It is also important to put oneself in the “user’s shoes.” This provides credibility when tough decisions have to be made about the project. Joshua found that by educating the primary stakeholder within the user group and gaining her buy-in on the project, he was then able to move outward to other stakeholders to get their buy-in on the project and their assistance throughout the requirements analysis phase of the project.

Furthermore, the software project manager needs to be a leader by being strong with the users. When the users ask for additional functionality, it is important to not simply agree to every request. A formal change request process is necessary to keep up with the changes and to ensure that the project will be ultimately successful. Two different project managers recalled a situation where they wished that they had simply walked away a project when users were being unreasonable. While this recourse is not always possible, several software project managers remarked on the importance of being strong with users that requested changes by educating them on the consequences of these decisions.

Trust

As in any situation where personal relationships are important, maintaining trust is a large component of managing user expectations.

If your client trusts you, they know you’re not trying to get something over on them... I believe that me and the client have to be friends. It’s always give and take. The client’s going to give, you’re going to give, you’re going to take, and he’s going to take. So it’s a relationship with each other. If you’re not in business with your client, then you’re going to pretty much be up the creek without a paddle. (Matthew)

At times, a software project manager may walk into a less than desirable situation. One project manager recounted a situation in which the users had selected a technology prior to determining the requirements. This meant that the software project manager had to force a solution even though there was a mismatch between the technology and requirements. The users were ultimately happy with the outcome after a lot of hard work. While some software project managers may feel the need to just make the solution work and hide these problems from the users, this project manager offered a different approach.

I mean as far as managing the client, the approach that we took was just being brutally honest with them and, typically that worked pretty well. (Joseph)

Even simple things, such as informing the users that they will see a prototype “in one week” rather than “soon,” was a tactic one project manager advocated to establish trust and ensure that mutual understanding between manager and user.

Users should hear about any problems on the project from the software project manager rather than someone else. Given the skeptical nature that some users have about software projects or any type of project that creates a significant impact on their work, communication with the users about the good and bad aspects of the project is key. One software project manager offered the following advice:

Communicate often, and make sure that your users hear of issues from you first before they hear it from elsewhere. The key is communication. You can create the worst system in the world, but if the user is happy with it, then you've got a chance of salvaging something. (Joshua)

Projects did go awry when the software project manager neglected to inform the client of problems on the project. Christopher recounted a situation in which he joined a project that was already under-scope. The users kept requesting more functionality for a project with too short of a timeline and too small of a budget. He was unwilling to talk to the client about the situation and believed he should hide the problems from the user. Ultimately the project was cancelled due to other problems at the firm; however, this unwillingness to disclose information about the project weakened the trust and relationship between this project manager and the users.

DISCUSSION

Implications for Practice

This research identified three broad strategies and many tactics to manage user expectations. This study found that failing to implement one or more strategies may result in a less than desirable outcome. Therefore, this paper provides a mini playbook of tactics that can be used by software project managers to counter this very common and very real risk that occurs on software projects.

Furthermore, by managing user expectations by using the strategies and tactics employed in this paper, several other risks could be addressed (at least partially). Risks, such as failure to gain user commitment, lack of frozen requirements, or changing scope/objectives (Schmidt et al. 2001), are addressed to a certain extent by using the strategies outlined in this work. Using these strategies to address several risks at once could help a software project manager to more successfully manage software projects.

Implications for Research

This research addresses one of the most common risks faced by software project managers, managing user expectations. The IS literature as well as marketing theories illustrate the importance managing user expectations to ensure future use and satisfaction with an information system. The current research can be extended further to determine if managing user expectations using these strategies does indeed lead to higher levels of user satisfaction of software once the system is deployed. Another clear extension of this research is to identify strategies and tactics to address other risks that are common across software projects.

CONCLUSION

Unfortunately, some consider the mantra for software development to be “over-promising and under-delivering.” Studies in the IS field have illustrated the importance of properly managing expectations and the role this has on user satisfaction and software use; however, the literature has failed to explain how to manage user expectations. This work addresses this void within the literature. This study identifies three general strategies and several specific tactics that software project managers can employ to counter this important risk on software projects.

The tactics identified by this research are not overly complex and are quite simple, yet if software project managers truly understood and followed this advice, managing user expectations would not be such a high ranking risk in software projects (Schmidt et al. 2001; Baccarini et al. 2004). Much of the traditional research on software project management neglects to ask project managers about their perspectives and ideas on how to manage projects (Rose et al. 2007). Therefore, I chose to solicit advice from those “in the trenches” who face the difficult task of managing user expectations on projects on a daily basis. Their lessons remind us of the importance of the user involvement, leadership, and trust on software projects.

REFERENCES

1. Baccarini, D., Salm, G. and Love, P.E.D. “Management of Risks in Information Technology Projects,” *Industrial Management & Data Systems* (104:4) 2004, pp 286-295.
2. Bhattacharjee, A. "Understanding Information Systems Continuance: An Expectation-Disconfirmation Model," *MIS Quarterly* (25:3), 2001, pp 351-370.
3. Bokhari, R.H. "The Relationship between System Usage and User Satisfaction: A Meta-Analysis," *The Journal of Enterprise Information Management* (18:2) 2005, pp 211-234.
4. Davis, F.D., and Venkatesh, V. "Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management," *IEEE Transactions on Engineering Management* (51) 2004, pp 31-46.

5. DeLone, W.H., and McLean, E.R. "Information Systems Success: The Quest for the Dependent Variable," *Information Systems Research* (3:1) 1992, pp 60-95.
6. DeLone, W.H., and McLean, E.R. "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update," *Journal of Management Information Systems* (19:4), 2003, pp 9-30.
7. Ginzberg, M.J. "Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions," *Management Science* (27:4) 1981, pp 459-478.
8. Kettinger, W.J., and Lee, C.C. "Perceived Service Quality and User Satisfaction with the Information Services Function," *Decision Sciences* (24:5/6) 1994, pp 737-766.
9. Kettinger, W.J., and Lee, C.C. "Zones of Tolerance: Alternative Scales for Measuring Information Systems Service Quality," *MIS Quarterly* (29:4), 1995, pp 607-621.
10. Kim, D.J., Ferrin, D.L., and Rao, H.R. "A Study of the Effect of Consumer Trust on Consumer Expectations and Satisfaction: The Korean Experience," *Proceedings of the 5th International Conference on Electronic Commerce*, ACM Press, Pittsburgh, PA, 2003, pp 310-315.
11. Mahmood, M.A., Brun, J.M., and Gemoets, L.A. "Variables Affecting Information Technology End-User Satisfaction: A Meta-Analysis of the Empirical Literature," *International Journal of Human-Computer Studies* (52) 2000, pp 751-771.
12. Oliver, R.L. "A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions," *Journal of Marketing Research* (17:3) 1980.
13. Parasuraman, A., Zeithaml, V.A., and Berry, L.L. "SERVQUAL: A Multiple-item Scale for Measuring Consumer Perceptions of Service Quality," *Journal of Retailing* (64:1), 1988, pp 12-40.
14. Pitt, L.F., Watson, R.T., and Kavan, C.B. "Service Quality: A Measure of Information Systems Effectiveness," *MIS Quarterly* (19:2), 1995, pp 173-187.
15. Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, Project Management Institute, Newtown Square, PA, 2004.
16. Rose, J., Pedersen, K., Hosbond, J.H., and Kraemmergaard, P. "Management Competences, Not Tools and Techniques: A Grounded Examination of Software Project Management at WM-Data," *Information and Software Technology* (49:6), 2007, pp 605-624.
17. Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems* (17:4), 2001, pp 5-36.
18. Staples, D.S., Wong, I., and Seddon, P.B. "Having Expectations of Information Systems Benefits That Matched Received Benefits: Does It Really Matter?," *Information & Management* (40:2), 2002, pp 115-131.
19. Tiwana, A. and Keil, M. "The One-Minute Risk Assessment Tool," *Communications of the ACM* (47:11), 2004, pp 73-77.